UCRL-JRNL-222227

Convergence Analysis of a Domain Decomposition Paradigm

R. E. Bank, P. S. Vassilevski

June 19, 2006

Computing and Visualization in Science

Updated October 14, 2003

Numerische Mathematik manuscript No.
(will be inserted by the editor)

Convergence Analysis of a Domain Decomposition Paradigm

Randolph E. Bank*, Panayot S. Vassilevski**

Bank: Department of Mathematics, University of California, San Diego, La Jolla, California 92093-0112. Email: rbank@ucsd.edu.
Vassilevski: Center for Applied Scientific Computation, Lawrence Livermore National Laboratory, Livermore CA 94550. Email: panayot@llnl.gov.

Received: June 12, 2006

Summary We describe a domain decomposition algorithm for use in several variants of the parallel adaptive meshing paradigm of Bank and Holst. This algorithm has low communication, makes extensive use of existing sequential solvers, and exploits in several important ways data generated as part of the adaptive meshing paradigm. We show that for an idealized version of the algorithm, the rate of convergence is independent of both the global problem size N and the number of subdomains p used in the domain decomposition partition. Numerical examples illustrate the effectiveness of the procedure.

Key words Domain decomposition, Bank–Holst algorithm, parallel adaptive grid generation

Mathematics Subject Classification (1991): 65N55, 65N50

1 Introduction

In [6,7], Bank and Holst introduced a general approach to parallel adaptive meshing for systems of elliptic partial differential equations.

* The work of this author was supported by the National Science Foundation under contract DMS-0511766. The UCSD Scicomp Beowulf cluster was built using funds provided by the National Science Foundation through SCREMS Grant 0112413, with matching funds from the University of California at San Diego.
** The work of this author was performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

This approach was motivated by the desire to keep communications costs low, and to allow sequential adaptive software (such as the software package PLTMG [3] used in this work) to be employed without extensive recoding.

The original paradigm has three main components:

Step I: Load Balancing. We solve a small problem on a coarse mesh, and use a posteriori error estimates to partition the mesh. Each subregion has approximately the same error, although subregions may vary considerably in terms of numbers of elements or gridpoints.

Step II: Adaptive Meshing. Each processor is provided the complete coarse mesh and instructed to sequentially solve the *entire* problem, with the stipulation that its adaptive refinement should be limited largely to its own partition. The target number of elements and grid points for each problem is the same. At the end of this step, the mesh is regularized such that the global mesh described in Step III is conforming.

Step III: Global Solve. The final global mesh consists of the union of the refined partitions provided by each processor. A final solution is computed using domain decomposition.

With this paradigm, the load balancing problem is reduced to the numerical solution of a small elliptic problem on a single processor, using a sequential adaptive solver such as PLTMG without requiring any modifications to the sequential solver. The bulk of the calculation in the adaptive meshing step also takes place independently on each processor and can also be performed with a sequential solver with no modifications necessary for communication. The only parts of the calculation requiring communication are:

- The initial fan-out of the mesh distribution to the processors at the beginning of Step II, once the decomposition is determined by the error estimator in load balancing.
- Mesh regularization at the end of Step II requires communication to produce a global conforming mesh.
- The domain decomposition solver in Step III requires communicating certain information about the interface system.

In [4], Bank considered a variant of the above approach in which the load balancing occurs on a much finer mesh. The motivation was to address some possible problems arising from the use of a coarse grid in computing the load balance. This variant also has three main components.

Step I: Load Balancing. On a single processor we adaptively create a *fine* mesh of size N_p, and use a posteriori error estimates to partition the mesh such that each subregion has approximately equal error, similar to Step I of the original paradigm.
Step II: Adaptive Meshing. Each processor is provided the complete adaptive mesh and instructed to sequentially solve the *entire* problem. However, in this case each processor should adaptively *coarsen* regions corresponding to other processors, and adaptively refine its own subregion. The size of the problem on each processor remains N_p, but this adaptive rezoning strategy concentrates the degrees of freedom in the processor's subregion. At the end of this step, the mesh is regularized such that the global mesh is conforming.
Step III: Global Solve. This step is the same as in the original paradigm.

Using the variant, the initial mesh can be of any size. Indeed, our choice of N_p is mainly for convenience and to simplify notation; any combination of coarsening and refinement could be allowed in Step II. Allowing the mesh in Step I to be finer increases the cost of both the solution and the load balance in Step I, but it allows flexibility in overcoming potential deficiencies of a very coarse mesh in the original paradigm. See [6,7,10] for numerical examples of the original paradigm and [4,5] for examples comparing the original and variant paradigms.

Although both the original paradigm and the variant use the same domain decomposition solver in Step III, the variant algorithm produced some unforeseen consequences. In particular, in the PLTMG package, in Step II of the paradigm, edges lying on the interface system can be refined as necessary. Vertices added during refinement steps can be deleted during coarsening steps, but the original vertices defining the interface system must remain in the mesh during Steps II and III of either paradigm. This restriction insures that the subdomains remain geometrically conforming across all processors, and also plays an important role in the mesh regularization algorithm applied at the end of Step II.

This point is of little consequence for the original paradigm because it is based mainly on refinement. However, it is quite significant for the variant. Indeed, for the variant, coarsening is limited to the *interiors* of subdomains corresponding to other processors, while the parts of the interface system lying in the coarse parts of the domain remain largely unchanged. Thus in the domain decomposition solver the local problem has an unusual structure, in that it is highly refined

on its own subdomain and its part of the interface system, it is very coarse in the interior of other subregions, and it has the original level of refinement on the coarse parts of the interface system.

The purpose of this work is to analyze the domain decomposition solver in the environment provided these parallel adaptive refinement strategies. For an idealized version of the algorithm we are able to show that the rate of convergence is independent of both N and p. See [28,16,26,32] for general background on domain decomposition methods. Some discussion of the method developed here and its predecessors can be found in [9,8,10,20,5], while related ideas in the multigrid context can be found in Mitchell [22–24].

Our analysis here is interesting for several reasons. First, the overall iteration does not have a symmetric error propagator, even in the case where the underlying continuous problem and its conforming finite element discretization are self-adjoint and positive definite. Thus we do not take an approach based on estimating generalized condition numbers, but rather make direct norm estimates for the error reduction. For a special case, we can frame the analysis in terms of a norm estimate for a symmetric, positive semidefinite matrix.

Second, while the approximate solution of the global problem belongs to a usual, globally conforming, finite element space, (in our case, continuous piecewise linear finite elements on a shape regular triangulation) the domain decomposition iteration itself is based on a saddle point formulation for nonconforming finite element spaces. The Lagrange multipliers, which are used to impose continuity at vertices along the interface, have the flavor of Dirac delta functions when viewed in the finite element context. An additional complication in the analysis arises from the fact that these Lagrange multipliers are not actually computed or updated as part of the iteration. Our saddle point formulation of the global system of equations can be viewed as a special (and very simple) example of a mortar element method [13, 12,15,31,30,19].

Another part of the analysis draws heavily upon interior estimates for finite element solutions. Such estimates have a long history; see for example the survey by Wahlbin [29]; see also [25,33]. Many of the techniques and much of the analysis are now quite standard. Our analysis also has some similarities to that of meshless methods [21, 1].

The remainder of the paper is organized as follows. In Section 2, we present our parallel domain decomposition solver using traditional finite element spaces and notation. In Section 3, we compute the error propagator using linear algebra and matrix/operator notation.

In Section 4 we provide norm estimates for the rate of convergence in a special case. These estimates are seen to be independent of N and p for the two-dimensional problems considered here. In Section 5, we discuss the practical implementation of the algorithm, in particular the derivation of the symmetric, positive definite systems that are solved on each processor and the parallel communication requirements. Finally, in Section 6 we provide some numerical results.

2 Preliminaries

Let $\Omega = \cup_{i=1}^{p} \Omega_i \subset \mathcal{R}^2$ denote the domain, decomposed into p geometrically conforming subdomains. Let Γ denote the interface system. The degree of a vertex x lying on Γ is the number of subregions for which $x \in \bar{\Omega}_i$. A cross point is a vertex $x \in \Gamma$ with degree$(x) \geq 3$. We assume that the maximal degree at cross points is bounded by the constant δ_0. The overlap of Ω_i is the number of other regions Ω_j for which $\bar{\Omega}_i \cap \bar{\Omega}_j \neq \emptyset$. We assume the overlap of Ω_i is bounded by the constant δ_1.

In this analysis, we will use several triangulations. The mesh $\mathcal{T}$ will be the globally refined, shape regular, quasiuniform, and conforming mesh of size h. We assume that the fine mesh $\mathcal{T}$ is aligned with the interface system Γ. There is a coarse mesh $\mathcal{T}^0$, also shape regular, conforming, and aligned with the interface system Γ. In the interior parts of the subdomains Ω_i, $1 \leq i \leq p$, the triangulation $\mathcal{T}^0$ is quasiuniform with elements of size $H \gg h$. To accommodate the variant paradigm, near the interface system Γ the mesh can be more refined. In particular, we will consider as a special case the situation where the fine interface system is completely represented in the mesh $\mathcal{T}^0$. To maintain shape regularity, the mesh $\mathcal{T}^0$ is graded in an appropriate (shape regular) fashion from the more refined elements near the interface Γ to the coarse elements of size H in the interiors of the Ω_i.

The triangulations $\mathcal{T}^i$, $1 \leq i \leq p$ are partially refined triangulations; they coincide with the fine triangulation $\mathcal{T}$ within Ω_i, but largely coincide with the coarse triangulation $\mathcal{T}^0$ elsewhere. We assume that the triangulations are nested in the following sense: for $1 \leq i \leq p$, we have $\mathcal{T}^0 \subset \mathcal{T}^i \subset \mathcal{T}$. The special case where the complete interface system is represented in $\mathcal{T}^0$ is the most simple situation. In particular, $\mathcal{T}^i$ exactly coincides with $\mathcal{T}$ in Ω_i, and exactly coincides with $\mathcal{T}^0$ in Ω_j, $j \neq i$.

In the case that the interface system is not represented in $\mathcal{T}^0$, $\mathcal{T}^i$ is a nonuniform refinement of $\mathcal{T}^0$, where the refinement is *mainly*

restricted to Ω_i. Since the interface system is coarse, edges in $\Gamma \cap \partial\Omega_i$ are refined, requiring some additional (graded) refinements outside of Ω_i in order to maintain conformity and shape regularity in the mesh. Thus, given two triangulations, $\mathcal{T}^i$ and $\mathcal{T}^j$, a course subdomain Ω_k, $k \neq i$, $k \neq j$, may be triangulated differently in the two cases, especially if Ω_k shares an interface with Ω_i or Ω_j.

Let $\mathcal{S}$ denote the space of piecewise linear polynomials, associated with the triangulation $\mathcal{T}$, that are continuous in each of the Ω_i, but can be discontinuous along the interface system Γ. Let $\bar{\mathcal{S}} \subset \mathcal{S}$ denote the subspace of globally continuous piecewise linear polynomials. The usual basis for $\mathcal{S}$ is just the union of the nodal basis functions corresponding to each of the subdomains Ω_i; such basis functions have their support in $\bar{\Omega}_i$ and those associated with nodes on Γ will have a jump at the interface. In the theory, we will have occasion to consider another basis, allowing us to write $\mathcal{S} = \bar{\mathcal{S}} \oplus \mathcal{X}$, where $\mathcal{X}$ is a subspace associated exclusively with jumps on Γ. In particular, we will use the global conforming nodal basis for the space $\bar{\mathcal{S}}$, and construct a basis for $\mathcal{X}$ as follows. Let z_k be a vertex lying on Γ shared by two regions Ω_i and Ω_j (for now, z_k is not a crosspoint). Let $\phi_{i,k}$ and $\phi_{j,k}$ denote the usual nodal basis functions corresponding to z_k in Ω_i and Ω_j, respectively. The continuous nodal basis function for z_k in $\bar{\mathcal{S}}$ is $\phi_k \equiv \phi_{i,k} + \phi_{j,k}$, and the "jump" basis function in $\mathcal{X}$ is $\hat{\phi}_k \equiv \phi_{i,k} - \phi_{j,k}$. The direction of the jump is arbitrary at each z_k, but once chosen, will be used consistently. In this example, at point z_k we will refer to i and the "master" index and j as the "slave" index. At a cross point where $\ell > 2$ subregions meet, there will be one nodal basis function corresponding to $\bar{\mathcal{S}}$ and $\ell - 1$ jump basis functions. These are constructed by choosing one master index for the point, and making the other $\ell - 1$ indices slaves. We can construct $\ell - 1$ basis functions for $\mathcal{X}$ as $\phi_{i,k} - \phi_{j,k}$, where i is the master index and j is one of the slave indices.

For each of the triangulations $\mathcal{T}^i$, $1 \leq i \leq p$, and for the global coarse triangulation $\mathcal{T}^0$, we have a global nonconforming subspace $\mathcal{S}^i \subset \mathcal{S}$, and global conforming subspace $\bar{\mathcal{S}}^i \subset \bar{\mathcal{S}}$. In a fashion similar to $\mathcal{S}$, we have $\mathcal{S}^i = \bar{\mathcal{S}}^i \oplus \mathcal{X}^i$. In the special case that $\mathcal{T}^0$ contains the globally refined interface system, $\mathcal{X}^i \equiv \mathcal{X}$, $1 \leq i \leq p$.

Let the continuous variational problem be: find $u \in \mathcal{H}^1(\Omega)$ such that

$$a(u, v) = (f, v) \tag{1}$$

for all $v \in \mathcal{H}^1(\Omega)$, where

$$a(u,v) = \int_\Omega a\nabla u \cdot \nabla v + buv\, dx,$$
$$(f,v) = \int_\Omega fv\, dx,$$
$$|\!|\!| u |\!|\!|_\Omega^2 = a(u,u).$$

We assume that $a > 0$, $b \geq 0$ are smooth and chosen such that $a(\cdot,\cdot)$ is coercive, so that $|\!|\!|\cdot|\!|\!|_\Omega$ defines a strong norm on $\mathcal{H}^1(\Omega)$, comparable to the usual $\|\cdot\|_{1,\Omega}$. The case of a singular Neumann problem presents no difficulties; the usual compatibility condition

$$(f,1) = 0$$

applies, and the solution is made unique by requiring

$$(u,1) = 0.$$

To deal with the nonconforming nature of $\mathcal{S}$, for $u, v \in \mathcal{S}$, we define

$$\begin{aligned} a(u,v) &= \sum_{i=1}^{p} \int_{\Omega_i} a\nabla u \cdot \nabla v + buv\, dx \\ &= \sum_{i=1}^{p} a_{\Omega_i}(u,v) \end{aligned}$$

For each vertex z lying on Γ there is one master index and $\ell - 1 > 0$ slave indices. The total number of slave indices is denoted by K, so the total number of constraint equations in our nonconforming method is K. To simplify notation, for each $1 \leq j \leq K$, let $m(j)$ denote the corresponding master index, and z_j the corresponding vertex. We define the bilinear form $b(v,\lambda)$ by

$$b(v,\lambda) = \sum_{j=1}^{K} \{v_{m(j)} - v_j\}\lambda_j \tag{2}$$

where $\lambda \in \mathcal{R}^K$. In words, $b(\cdot,\cdot)$ measures the jump between the master value and each of the slave values at each vertex on Γ. The nonconforming variational formulation of (1) is: find $u_h \in \mathcal{S}$ such that

$$\begin{aligned} a(u_h,v) + b(v,\lambda) &= (f,v) \\ b(u_h,\xi) \qquad\quad &= 0 \end{aligned} \tag{3}$$

for all $v \in \mathcal{S}$ and $\xi \in \mathcal{R}^K$. Although this is formally a saddle point problem, the constraints are very simple; in particular, (3) simply imposes continuity at each of the vertices lying on Γ, which in turn, implies that $u_h \in \bar{\mathcal{S}}$. Thus u_h also solves the reduced and conforming variational problem: find $u_h \in \bar{\mathcal{S}}$ such that

$$a(u_h, v) = (f, v) \tag{4}$$

for all $v \in \bar{\mathcal{S}}$.

In the triangulations $\mathcal{T}^i$, the mesh near Γ may not be as refined as in $\mathcal{T}$ in Ω_j, $j \neq i$. Let $\mathcal{K}_i$ denote the index set of constraint equations in (2) that correspond to vertices present in $\mathcal{T}^i$. Then

$$b_i(v, \lambda) = \sum_{j \in \mathcal{K}_i} \{v_{m(j)} - v_j\} \lambda_j.$$

If the interface is completely refined $b_i(\cdot, \cdot) \equiv b(\cdot, \cdot)$.

We are now in a position to formulate our domain decomposition algorithm. We first consider the initial guess $u_0 \in \mathcal{S}$, generated as follows: for $1 \leq i \leq p$, we find (in parallel) $u_{0,i} \in \bar{\mathcal{S}}^i$ satisfying

$$a(u_{0,i}, v) = (f, v) \tag{5}$$

for all $v \in \bar{\mathcal{S}}^i$. Note that here we assume exact solution of these local problems; in the actual implementation, these are solved approximately using the multigraph algorithm. The initial guess $u_0 \in \mathcal{S}$ is composed by taking the part of $u_{0,i}$ corresponding to the fine subregion Ω_i for each i. In particular, let χ_i be the characteristic function for the subregion Ω_i. Then

$$u_0 = \sum_{i=1}^{p} \chi_i u_{0,i} \tag{6}$$

To compute $u_{k+1} \in \mathcal{S}$ from $u_k \in \mathcal{S}$, we solve (in parallel): for $1 \leq i \leq p$, find $e_{k,i} \in \mathcal{S}^i$ and $\lambda_{k,i} \in \mathcal{R}^K$ such that

$$\begin{aligned} a(e_{k,i}, v) + b_i(v, \lambda_{k,i}) &= (f, v) - a(u_k, v) \\ b_i(e_{k,i}, \xi) \qquad\qquad &= -b_i(u_k, \xi) \end{aligned} \tag{7}$$

for all $v \in \mathcal{S}^i$ and $\xi \in \mathcal{R}^K$. We then form

$$\begin{aligned} e_k &= \sum_{i=1}^{p} \chi_i e_{k,i}, \\ u_{k+1} &= u_k + e_k. \end{aligned} \tag{8}$$

We pause to make a few remarks. First, although the u_k and e_k are elements of the nonconforming space $\mathcal{S}$, the limit function $u_\infty = u_h$ belongs to the conforming finite element space $\bar{\mathcal{S}}$. In some sense, the purpose of the iteration is to drive the jumps in the approximate solution u_k to zero. Second, although (7) suggests a saddle point problem needs to be solved, by recognizing that only $\chi_i e_{k,i}$ is actually used, one can reduce (7) to a positive definite problem of the form (5). In particular, the Lagrange multipliers $\lambda_{k,i}$ need not be computed or updated. This aspect is described in detail in Section 5.

Let $\hat{e}_k = u_h - u_k$ denote the exact error in u_k. Then $\hat{e}_k$ satisfies the saddle point problem: find $\hat{e}_k \in \mathcal{S}$ and $\lambda_k \in \mathcal{R}^K$ such that

$$\begin{aligned} a(\hat{e}_k, v) + b(v, \lambda_k) &= (f, v) - a(u_k, v) \\ b(\hat{e}_k, \xi) \qquad\quad &= -b(u_k, \xi) \end{aligned} \tag{9}$$

for all $v \in \mathcal{S}$ and $\xi \in \mathcal{R}^K$.

By comparing (7) and (9), we see that

$$\begin{aligned} a(\hat{e}_k - e_{k,i}, v) + b_i(v, \lambda_k - \lambda_{k,i}) &= 0 \\ b_i(\hat{e}_k - e_{k,i}, \xi) \qquad\quad &= 0 \end{aligned} \tag{10}$$

for all $v \in \mathcal{S}^i$ and $\xi \in \mathcal{R}^K$. For the special case where $b_i(\cdot,\cdot) \equiv b(\cdot,\cdot)$ we also have the more simple projection-like relation

$$a(\hat{e}_k - e_{k,i}, v) = 0 \tag{11}$$

for all $v \in \bar{\mathcal{S}}^i$. From the identity

$$v = \sum_{i=1}^{p} \chi_i v$$

for all $v \in \mathcal{S}$, we have

$$\hat{e}_{k+1} = \hat{e}_k - e_k = \sum_{i=1}^{p} \chi_i (\hat{e}_k - e_{k,i}). \tag{12}$$

3 Derivation of the Error Propagator

In this section we will derive, in matrix/operator notation, the error propagator for the iteration described in Section 2. We will begin for the simple case of $p = 2$ subregions. We then generalize to the case of general p, but with a completely refined interface system on each processor (i.e., $b_i(\cdot,\cdot) \equiv b(\cdot,\cdot)$). Finally we consider general p, with a coarse interface system in the region outside of a given processors' subdomain.

3.1 The case $p = 2$

In the case $p = 2$, the global mortar matrix is given by

$$A = \begin{pmatrix} A_1 & 0 & B_1^t \\ 0 & A_2 & B_2^t \\ B_1 & B_2 & 0 \end{pmatrix}. \tag{13}$$

The matrices A_i correspond to the bilinear forms $a_{\Omega_i}(\cdot,\cdot)$ relative to the global fine mesh $\mathcal{T}$. The matrices B_i correspond to the bilinear form $b(\cdot,\cdot)$; these matrices are rectangular with one nonzero entry (± 1) for each row that corresponds to a constraint equation on $\partial\Omega_i$; this is all rows for the case $p = 2$, but in general many rows of B_i will contain all zeroes. The global Schur complement is given by

$$S = B_1 A_1^{-1} B_1^t + B_2 A_2^{-1} B_2^t.$$

The restriction operator mapping $\mathcal{S} \to \mathcal{S}^1$ is:

$$Q_1 = \begin{pmatrix} I & 0 & 0 \\ 0 & P_2^t & 0 \\ 0 & 0 & I \end{pmatrix}. \tag{14}$$

The partition of unity matrix for region one is

$$\chi_1 = \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{15}$$

Let

$$\begin{aligned} \bar{A}_2 &= P_2^t A_2 P_2 \\ \bar{B}_2 &= B_2 P_2 \\ \pi_2 &= I - P_2 \bar{A}_2^{-1} P_2^t A_2 \\ S_1 &= B_1 A_1^{-1} B_1^t + \bar{B}_2 \bar{A}_2^{-1} \bar{B}_2^t \\ &= B_1 A_1^{-1} B_1^t + B_2 (I - \pi_2) A_2^{-1} B_2^t \\ E_1 &= A_1^{-1} B_1^t S_1^{-1}. \end{aligned}$$

Note that the elliptic projection π_2 removes low frequency components from region two, whereas the extension operator E_1 makes smooth extension of the data restricted to the interface into region one. As usual, if A_i is singular, A_i^{-1} is understood in the sense of generalized inverse. In our situation, A_i might have a one dimensional kernel corresponding to a constant.

The subdomain solver for region one is

$$M_1 = Q_1 A Q_1^t = \begin{pmatrix} A_1 & 0 & B_1^t \\ 0 & \bar{A}_2 & \bar{B}_2^t \\ B_1 & \bar{B}_2 & 0 \end{pmatrix}$$
$$= \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ B_1 A_1^{-1} & \bar{B}_2 \bar{A}_2^{-1} & I \end{pmatrix} \begin{pmatrix} A_1 & 0 & 0 \\ 0 & \bar{A}_2 & 0 \\ 0 & 0 & -S_1 \end{pmatrix} \begin{pmatrix} I & 0 & A_1^{-1} B_1^t \\ 0 & I & \bar{A}_2^{-1} \bar{B}_2^t \\ 0 & 0 & I \end{pmatrix}$$

Using the factorization, it is easy to compute

$$\chi_1 M_1^{-1} Q_1 A = \begin{pmatrix} I & E_1 B_2 \pi_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{16}$$

A similar calculation holds for region two. Let

$$Q_2 = \begin{pmatrix} P_1^t & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}$$
$$\chi_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
$$M_2 = Q_2 A Q_2^t = \begin{pmatrix} \bar{A}_1 & 0 & \bar{B}_1^t \\ 0 & A_2 & B_2^t \\ \bar{B}_1 & B_2 & 0 \end{pmatrix}$$
$$\bar{A}_1 = P_1^t A_1 P_1$$
$$\bar{B}_1 = B_1 P_1$$
$$\pi_1 = I - P_1 \bar{A}_1^{-1} P_1^t A_1$$
$$S_2 = \bar{B}_1 \bar{A}_1^{-1} \bar{B}_1^t + B_2 A_2^{-1} B_2^t$$
$$= B_1 (I - \pi_1) A_1^{-1} B_1^t + B_2 A_2^{-1} B_2^t$$
$$E_2 = A_2^{-1} B_2^t S_2^{-1}.$$

Similar to region one, we have

$$\chi_2 M_2^{-1} Q_2 A = \begin{pmatrix} 0 & 0 & 0 \\ E_2 B_1 \pi_1 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{17}$$

Using (16)-(17), the global error propagator is given by

$$G = I - \chi_1 M_1^{-1} Q_1 A - \chi_2 M_2^{-1} Q_2 A$$
$$= - \begin{pmatrix} 0 & E_1 B_2 \pi_2 & 0 \\ E_2 B_1 \pi_1 & 0 & 0 \\ 0 & 0 & -I \end{pmatrix}. \tag{18}$$

The I in the $(3,3)$ block arises because we do not compute or update the Lagrange multipliers. The fact that the block third row and column are otherwise zero shows that failing to compute Lagrange multipliers does not affect the error in the other components of the solution. Thus, since we are only interested in the error in the solution itself, it suffices to consider on the block 2×2 error propagator

$$\bar{G} = -\begin{pmatrix} 0 & E_1 B_2 \pi_2 \\ E_2 B_1 \pi_1 & 0 \end{pmatrix}. \tag{19}$$

We note that $\bar{G}$ is not a symmetric operator (including the energy norm).

3.2 The case of general p

We now consider the case of general p, assuming that the global interface system is completely represented on all processors. In analogy with (13), the global mortar matrix is given by

$$A = \begin{pmatrix} A_1 & 0 & \dots & 0 & B_1^t \\ 0 & A_2 & & 0 & B_2^t \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & A_p & B_p^t \\ B_1 & B_2 & \dots & B_p & 0 \end{pmatrix}. \tag{20}$$

The global Schur complement is given by

$$S = \sum_{j=1}^{p} B_j A_j^{-1} B_j^t.$$

We will derive the contribution to the global error propagator associated with region one. The remaining regions follow a similar pattern. In this context, it is useful to consider regions $2-p$ as a single block and express the global matrix as

$$A = \begin{pmatrix} A_1 & 0 & B_1^t \\ 0 & A_* & B_*^t \\ B_1 & B_* & 0 \end{pmatrix}. \tag{21}$$

Note A_* is a block diagonal matrix, and B_*^t is a block vector. We can now follow the derivation for the case $p = 2$; in analogy with (14), the restriction operator for region one is:

$$Q_1 = \begin{pmatrix} I & 0 & 0 \\ 0 & P_*^t & 0 \\ 0 & 0 & I \end{pmatrix}, \tag{22}$$

where P_* is a block diagonal matrix. The partition of unity matrix for region one is given by (15). The subdomain solver for region one is

$$M_1 = Q_1 A Q_1^t = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ B_1 A_1^{-1} & \bar{B}_* \bar{A}_*^{-1} & I \end{pmatrix} \begin{pmatrix} A_1 & 0 & 0 \\ 0 & \bar{A}_* & 0 \\ 0 & 0 & -S_1 \end{pmatrix} \begin{pmatrix} I & 0 & A_1^{-1} B_1^t \\ 0 & I & \bar{A}_*^{-1} \bar{B}_*^t \\ 0 & 0 & I \end{pmatrix}$$

where

$$\begin{aligned} \bar{A}_* &= P_*^t A_* P_* \\ \bar{B}_* &= B_* P_* \\ \pi_* &= I - P_* \bar{A}_*^{-1} P_*^t A_* \\ S_1 &= B_1 A_1^{-1} B_1^t + \bar{B}_* \bar{A}_*^{-1} \bar{B}_*^t \\ &= B_1 A_1^{-1} B_1^t + \sum_{j=2}^{p} \bar{B}_j \bar{A}_j^{-1} \bar{B}_j^t \\ E_1 &= A_1^{-1} B_1^t S_1^{-1}. \end{aligned}$$

Note in particular that π_* is block diagonal. Using the factorization, and following the pattern for the case $p = 2$ it is easy to compute

$$\chi_1 M_1^{-1} Q_1 A = \begin{pmatrix} I & E_1 B_* \pi_* & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

We can express this using the expanded block structure as

$$\chi_1 M_1^{-1} Q_1 A = \begin{pmatrix} I & E_1 B_2 \pi_2 & \dots & E_1 B_p \pi_p & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}. \tag{23}$$

After making a similar calculation for the remaining regions, we can compute the global error propagator given by

$$\begin{aligned} G &= I - \sum_{j=1}^{p} \chi_j M_j^{-1} Q_j A \\ &= - \begin{pmatrix} 0 & E_1 B_2 \pi_2 & \dots & E_1 B_p \pi_p & 0 \\ E_2 B_1 \pi_1 & 0 & & E_2 B_p \pi_p & 0 \\ \vdots & & \ddots & & \vdots \\ E_p B_1 \pi_1 & E_p B_2 \pi_2 & & 0 & 0 \\ 0 & 0 & \dots & 0 & -I \end{pmatrix}. \end{aligned} \tag{24}$$

As before the appearance of the identity in the last row corresponds to the fact that we do not compute the Lagrange multipliers as part of the iteration. As in the case $p = 2$, we can restrict attention to the block $p \times p$ matrix

$$\bar{G} = -\begin{pmatrix} 0 & E_1 B_2 \pi_2 & \dots & E_1 B_p \pi_p \\ E_2 B_1 \pi_1 & 0 & & E_2 B_p \pi_p \\ \vdots & & \ddots & \\ E_p B_1 \pi_1 & E_p B_2 \pi_2 & & 0 \end{pmatrix} \tag{25}$$

We note that $\bar{G}$ can be factored as

$$\bar{G} = -\bar{E}\bar{\mathcal{J}}\bar{\pi} \tag{26}$$

where

$$\bar{E} = \begin{pmatrix} E_1 & & & \\ & E_2 & & \\ & & \ddots & \\ & & & E_p \end{pmatrix},$$

$$\bar{\pi} = \begin{pmatrix} B_1\pi_1 & & & \\ & B_2\pi_2 & & \\ & & \ddots & \\ & & & B_p\pi_p \end{pmatrix},$$

$$\bar{\mathcal{J}} = \begin{pmatrix} 0 & \mathcal{J}_2 & \dots & \mathcal{J}_p \\ \mathcal{J}_1 & 0 & & \mathcal{J}_p \\ \vdots & & \ddots & \\ \mathcal{J}_1 & \dots & \mathcal{J}_{p-1} & 0 \end{pmatrix}.$$

Here $\mathcal{J}_i$ is a diagonal matrix with zeros and ones on the diagonal. A diagonal entry is one if the corresponding constraint equation involves a point on the interface of region i, and is zero otherwise. In particular, note that $\mathcal{J}_i B_i = B_i$.

Let

$$\begin{aligned} F_i &= B_i \pi_i A_i^{-1} B_i^t \\ S_0 &= \sum_{j=1}^{p} \bar{B}_j \bar{A}_j^{-1} \bar{B}_j^t = S_i - F_i \\ \hat{F}_i &= S_0^{-1/2} F_i S_0^{-1/2} \end{aligned} \tag{27}$$

Then

$$B_i \pi_i A_i^{-1} B_i^t S_i^{-1} = F_i (S_0 + F_i)^{-1} = S_0^{1/2} \hat{F}_i (I + \hat{F}_i)^{-1} S_0^{-1/2}$$

We note that S_0, the Schur complement for the coarse space $\mathcal{S}^0$, is symmetric and positive definite (taking into account possible constraints for the case of a singular Neumann problem), and $\hat{F}_i(I+\hat{F}_i)^{-1}$ is symmetric and positive semidefinite.

Let

$$D = \mathrm{diag}(\mathcal{J}_i),$$
$$V^t = \left(\mathcal{J}_1\ \mathcal{J}_2\ \ldots\ \mathcal{J}_p\right),$$
$$W^t = \left(I\ I\ \ldots\ I\right).$$

Then

$$\bar{\mathcal{J}} = WV^t - D.$$

We also note that (since $\mathcal{J}_i B_i = B_i$)

$$\bar{G} = -\bar{E}\bar{\mathcal{J}}\bar{\pi} = -\bar{E}(WW^t - I)\bar{\pi}.$$

Let

$$T = \mathrm{diag}\left(\left\{\hat{F}_i(I+\hat{F}_i)^{-1}\right\}^{1/2}\right).$$

Then

$$-(WW^t - I)\bar{\pi}\bar{E} = -\mathrm{diag}(S_0^{1/2})(WW^t - I)T^2\mathrm{diag}(S_0^{-1/2}).$$

Thus the analysis of the error propagator $\bar{G}$ can be reduced to analyzing the *symmetric* error propagator

$$\hat{G} = -T(WW^t - I)T. \tag{28}$$

3.3 The case of general p with coarsened interface

We now assume that the global interface system is coarsened on each processor. Our goal is to determined the changes relative to the case of the globally refined interface system considered in the last subsection. Rather than make a complete derivation, we will concentrate on the differences. As before, we will consider just the contribution to the global error propagator due to processor one.

Generalizing (20), the restriction operator for region one now is defined by:

$$Q_1 = \begin{pmatrix} I & 0 & 0 \\ 0 & P_*^t & 0 \\ 0 & 0 & R_1 \end{pmatrix}, \tag{29}$$

where P_* is a block diagonal matrix as before, and R_1 is a restriction matrix for the Lagrange multipliers. R_1 is rectangular; each row

has a single entry of 1 and the remaining entries are 0; essentially R_1 selects the constraint equations to be imposed on processor one; note that this is the set of all constraint equations for interface grid points present in $\mathcal{T}^1$, and in particular, all constraint equations for the interface boundary of Ω_1 in $\mathcal{T}$.

The partition of unity matrix for region one is still given by (15). Let

$$\begin{aligned}\bar{S}_1 &= R_1 S_1 R_1^t \\ C_1 &= I - R_1^t \bar{S}_1^{-1} R_1 S_1 \\ \bar{E}_1 &= A_1^{-1} B_1^t R_1^t \bar{S}_1^{-1} R_1.\end{aligned}$$

S_1 is the Schur complement, defined previously, obtained if all the interface constraints are imposed on processor one; because of the structure of R_1, $\bar{S}_1$ is a submatrix of S_1 corresponding to the constraint equations which are imposed on processor one. C_1 is the corresponding projection matrix. Note that now $\bar{E}_1$ is defined using the restricted Schur complement; if all constraints are imposed, $R_1 = I$, and $\bar{E}_1 = E_1$.

Following the pattern of the previous subsections, it is straightforward to see, in analogy to (23),

$$\chi_1 M_1^{-1} Q_1 A = \begin{pmatrix} I & \bar{E}_1 B_2 \pi_{12} & \dots & \bar{E}_1 B_p \pi_{1p} & A_1^{-1} B_1^t C_1 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}. \tag{30}$$

The global error propagator is given by

$$\begin{aligned} G &= I - \sum_{j=1}^{p} \chi_j M_j^{-1} Q_j A \\ &= - \begin{pmatrix} 0 & \bar{E}_1 B_2 \pi_{12} & \dots & \bar{E}_1 B_p \pi_{1p} & A_1^{-1} B_1^t C_1 \\ \bar{E}_2 B_1 \pi_{21} & 0 & & \bar{E}_2 B_p \pi_{2p} & A_2^{-1} B_2^t C_2 \\ \vdots & & \ddots & & \vdots \\ \bar{E}_p B_1 \pi_{p1} & \bar{E}_p B_2 \pi_{p2} & & 0 & A_p^{-1} B_p^t C_p \\ 0 & 0 & \dots & 0 & -I \end{pmatrix}. \end{aligned} \tag{31}$$

As before the appearance of the identity in the last row corresponds to the fact that we do not compute the Lagrange multipliers as part of the iteration. However, the last block column of G is not otherwise zero as in the previous case (24). Thus the fact that not

all constraints are imposed on all processors does influence the convergence of the overall iteration. However, because G remains block upper triangular, this block column does not effect the eigenvalues of the iteration matrix and hence is asymptotically benign. So, as in the previous cases, we can restrict attention to the block $p \times p$ matrix

$$\bar{G} = - \begin{pmatrix} 0 & \bar{E}_1 B_2 \pi_{12} & \dots & \bar{E}_1 B_p \pi_{1p} \\ \bar{E}_2 B_1 \pi_{21} & 0 & & \bar{E}_2 B_p \pi_{2p} \\ \vdots & & \ddots & \\ \bar{E}_p B_1 \pi_{p1} & \bar{E}_p B_2 \pi_{p2} & & 0 \end{pmatrix}. \tag{32}$$

A more serious problem arises in the local projectors, now denoted π_{ij}. The extra subscript is needed because, e.g., the coarse triangulation of Ω_2 on processor one may now be different from the course triangulation of Ω_2 on processor three; hence $\pi_{12} \neq \pi_{32}$.

Because of this, $\bar{G}$ can no longer be factored as in (26). However, we still have the factorization

$$\bar{G} = -\bar{E}\bar{\Pi} \tag{33}$$

where

$$\bar{E} = \begin{pmatrix} \bar{E}_1 & & & \\ & \bar{E}_2 & & \\ & & \ddots & \\ & & & \bar{E}_p \end{pmatrix},$$

$$\bar{\Pi} = \begin{pmatrix} 0 & B_2 \pi_{12} & \dots & B_p \pi_{1p} \\ B_1 \pi_{21} & 0 & & B_p \pi_{2p} \\ \vdots & & \ddots & \\ B_1 \pi_{p1} & \dots & B_{p-1} \pi_{pp-1} & 0 \end{pmatrix}.$$

4 Convergence Analysis for a Special Case

Our goal is to analyze the error propagator $\bar{G} = -\bar{E}\bar{\mathcal{J}}\bar{\pi}$ in (26). Since $\bar{G}^k$ is the relevant operator for k iterations, it is sufficient to consider the operator $\bar{\pi}\bar{E}\bar{\mathcal{J}}$. Notice that $\bar{\pi}\bar{E}$ is a block diagonal (local) operator. In particular, $B_i \pi_i A_i^{-1} B_i^t S_i^{-1}$ behaves as follows. We begin with some function values (errors) defined on the global interface system. $B_i^t S_i^{-1}$ takes this global system of interface errors and maps it into (discrete) Neumann data for subregion i, which is then extended (discrete harmonic) by A_i^{-1} to all of $\bar{\Omega}_i$. We then apply the coarse grid projection π_i to this smooth extension; this is very much like a

multigrid coarse grid correction, and its effect on the smooth error is quite similar. The remaining (non-smooth) error after the projection is then restricted to the interface by B_i. The "mixing" matrix $\bar{\mathcal{J}}$ represents the global part of the iteration; $\bar{\mathcal{J}}$ takes the interface errors from region i and broadcasts them to all other processors; region i in turn receives similar errors from all other processors, and the local part of the cycle is repeated. Below we treat this process in the special case of a fully refined interface by analyzing the symmetric error propagator $\hat{G}$ of (28).

4.1 Some local error estimates

Let Ω_i be one of the domains. Let $\mathcal{S}_h \equiv \mathcal{S}_h(\Omega_i)$ denote the fine subspace on Ω_i (equivalently, the restriction of any of the global spaces $\mathcal{S}$, $\bar{\mathcal{S}}$, $\mathcal{S}^i$, or $\bar{\mathcal{S}}^i$ to Ω_i). Let $\mathcal{S}_H \equiv \mathcal{S}_H(\Omega_i) \subset \mathcal{S}_h$ denote the partially coarsened space (restriction of the global spaces $\mathcal{S}^0$, $\bar{\mathcal{S}}^0$, $\mathcal{S}^j$, or $\bar{\mathcal{S}}^j$, $j \neq i$, to Ω_i). Near $\partial\Omega_i \setminus \partial\Omega$ there is a strip of width at least d where $\mathcal{S}_h$ and $\mathcal{S}_H$ exactly coincide. Let Ω_d denote the interior of Ω_i where the two spaces differ, and let $\Omega_d \subset \Omega_{\hat{d}} \subset \Omega_i$. Informally, $\partial\Omega_{\hat{d}}$ lies halfway between $\partial\Omega_d$ and $\partial\Omega_i$. More precisely, along $\partial\Omega_i \setminus \partial\Omega$, the distance from $\partial\Omega_{\hat{d}}$ to both $\partial\Omega_d$ and $\partial\Omega_i$ is of order $d/2$.

We consider three Neumann problems: First: find $u \in \mathcal{H}^1(\Omega_i)$ such that

$$a_{\Omega_i}(u, v) = \langle g, v\rangle_{\partial\Omega_i}$$

for all $v \in \mathcal{H}^1(\Omega_i)$. Second: find $u_h \in \mathcal{S}_h$ such that

$$a_{\Omega_i}(u_h, v) = \langle g, v\rangle_{\partial\Omega_i}$$

for all $v \in \mathcal{S}_h$. Third: find $u_H \in \mathcal{S}_H$ such that

$$a_{\Omega_i}(u_H, v) = \langle g, v\rangle_{\partial\Omega_i}$$

for all $v \in \mathcal{S}_H$. The Neumann data g is a finite element function from $\mathcal{S}_h$ (or equivalently, $\mathcal{S}_H$) restricted to $\partial\Omega_i$. We assume unique solutions to these problems, with the usual caveats of consistency ($\langle g, 1\rangle_{\partial\Omega_i} = 0$) and unique generalized solutions ($(u, 1)_{\Omega_i} = (u_h, 1)_{\Omega_i} = (u_H, 1)_{\Omega_i} = 0$) for singular Neumann problems.

Since $\mathcal{S}_H \subset \mathcal{S}_h \subset \mathcal{H}^1(\Omega_i)$, we have the usual orthogonality relations

$$\begin{aligned} a_{\Omega_i}(u - u_h, v) &= 0 \quad \text{for all } v \in \mathcal{S}_h, \\ a_{\Omega_i}(u - u_H, v) &= 0 \quad \text{for all } v \in \mathcal{S}_H, \\ a_{\Omega_i}(u_h - u_H, v) &= 0 \quad \text{for all } v \in \mathcal{S}_H. \end{aligned}$$

We begin by showing that these three functions all have approximately the same energy.

Theorem 1 *Let u, u_h and u_H be defined as above. Then*

$$\|\!|u_h|\!\|_{\Omega_i} \le \|\!|u|\!\|_{\Omega_i} \le C\|\!|u_h|\!\|_{\Omega_i} \tag{34}$$

$$\|\!|u_H|\!\|_{\Omega_i} \le \|\!|u|\!\|_{\Omega_i} \le C\|\!|u_H|\!\|_{\Omega_i} \tag{35}$$

$$\|\!|u_H|\!\|_{\Omega_i} \le \|\!|u_h|\!\|_{\Omega_i} \le C\|\!|u_H|\!\|_{\Omega_i}. \tag{36}$$

Proof From the orthogonality relations we have

$$\|\!|u|\!\|^2_{\Omega_i} = \|\!|u_h|\!\|^2_{\Omega_i} + \|\!|u-u_h|\!\|^2_{\Omega_i}.$$

The left inequality of (34) is an immediate consequence. The left inequalities of (35)–(36) follow in a similar fashion. The right hand inequality in (34) holds because g is the trace of a finite element function $\psi \in S_h$. Let $q_h : L_2(\partial\Omega_i) \mapsto S_h|_{\partial\Omega_i}$ be the L_2–projection. Then, for any $\varphi \in H^1(\Omega_i)$ one has,

$$a_{\Omega_i}(u,\varphi) = \langle g,\varphi\rangle_{\partial\Omega_i} = \langle g, q_h\varphi\rangle_{\partial\Omega_i} = a_{\Omega_i}(u_h, \widetilde{(q_h\varphi)}),$$

where $\widetilde{(q_h\varphi)} \in S_h$ is any extension of $q_h\varphi$ to a function in S_h to the interior of Ω_i. Therefore,

$$\|\!|u|\!\|_{\Omega_i} = \sup_{\varphi\in H^1(\Omega_i)} \frac{a_{\Omega_i}(u_h, \widetilde{(q_h\varphi)})}{\|\!|\varphi|\!\|_{\Omega_i}} \le \|\!|u_h|\!\|_{\Omega_i} \sup_{\varphi\in H^1(\Omega_i)} \frac{\|\!|\widetilde{(q_h\varphi)}|\!\|_{\Omega_i}}{\|\!|\varphi|\!\|_{\Omega_i}}.$$

The extension $\widetilde{(q_h\varphi)}$ of the trace $q_h\varphi$ can be chosen so that it is bounded in energy, [28], i.e.,

$$\|\!|\widetilde{(q_h\varphi)}|\!\|_{\Omega_i} \le C\|q_h\varphi\|_{\frac12,\partial\Omega_i}.$$

By a trace inequality

$$\|\varphi\|_{\frac12,\partial\Omega_i} \le C\|\varphi\|_{1,\Omega_i} \le C\|\!|\varphi|\!\|_{\Omega_i}.$$

Thus we obtain

$$\|\!|u|\!\|_{\Omega_i} \le C\|\!|u_h|\!\|_{\Omega_i} \sup_{\varphi\in H^{\frac12}(\partial\Omega_i)} \frac{\|q_h\varphi\|_{\frac12,\partial\Omega_i}}{\|\varphi\|_{\frac12,\partial\Omega_i}}.$$

The right inequality in (34) follows since the L_2–projection q_h is bounded, i.e.,

$$\|q_h\varphi\|_{\frac12,\partial\Omega_i} \le C\|\varphi\|_{\frac12,\partial\Omega_i}.$$

The right hand inequality of (35) follows from an analogous argument. The right hand inequality in (36) follows from (34)–(35),

$$\|\!|u_h|\!\|_{\Omega_i} \le \|\!|u|\!\|_{\Omega_i} \le C\|\!|u_H|\!\|_{\Omega_i}.$$

The central error estimate is:

Theorem 2 *We have*

$$\|\!|u_h - u_H|\!\|_{\Omega_i} \leq C\frac{H}{d}\|\!|u_h|\!\|_{\Omega_i}. \tag{37}$$

Proof Let $\mathcal{I}_H$ denote an extension operator from S_h restricted to $\Omega_i \setminus \Omega_d$ into $S_H(\Omega_d)$. We assume that $\mathcal{I}_H$ is bounded, i.e., that the following estimate holds,

$$\|\!|\mathcal{I}_H v|\!\|_{\Omega_d} \leq C \inf_{w\in H^1(\Omega_d):\, w|_{\Omega_i\setminus\Omega_d} = v|_{\Omega_i\setminus\Omega_d}} \|\!|w|\!\|_{\Omega_d}. \tag{38}$$

The latter estimate represents the fact that $\mathcal{I}_H$ has a norm comparable with the minimum–norm continuous extension. For some explicit extension operators, see, for example [18]. In particular, it follows from (38) that

$$\|\!|\mathcal{I}_H(u_h - u_H)|\!\|_{\Omega_d} \leq C\|\!|u_h - u_H|\!\|_{\Omega_d}. \tag{39}$$

Using the fact that for all $v \in \mathcal{S}_h$, $v - \mathcal{I}_H v \equiv 0$ in $\Omega_i \setminus \Omega_d$, we have

$$\begin{aligned}
\|\!|u_h - u_H|\!\|^2_{\Omega_i} &= a_{\Omega_i}(u_h - u_H, u_h - u_H) \\
&= a_{\Omega_i}(u_h - u_H, u_h - u_H - \mathcal{I}_H(u_h - u_H)) \\
&= a_{\Omega_d}(u_h - u_H, u_h - u_H - \mathcal{I}_H(u_h - u_H))
\end{aligned}$$

where $a_{\Omega_d}(\cdot,\cdot)$ denotes the restriction of $a_{\Omega_i}(\cdot,\cdot)$ to Ω_d. Thus

$$\|\!|u_h - u_H|\!\|_{\Omega_i} \leq C\left\{\|\!|u_h - u_H|\!\|_{\Omega_d} + \|\!|\mathcal{I}_H(u_h - u_H)|\!\|_{\Omega_d}\right\}$$

Based on the boundedness of $\mathcal{I}_H$ one obtains

$$\|\!|u_h - u_H|\!\|_{\Omega_i} \leq C\|\!|u_h - u_H|\!\|_{\Omega_d}. \tag{40}$$

We now treat the right hand side of (40) using interior estimates. Let $\Omega_d \subset \Omega_{\hat{d}} \subset \Omega_i$ as described above. Then standard interior estimates for $\|\!|u - u_h|\!\|_{\Omega_d}$ [29,25] yield

$$\|\!|u - u_h|\!\|_{\Omega_d} \leq C\left\{\|\!|u - \chi|\!\|_{\Omega_{\hat{d}}} + d^{-1}\|u - u_h\|_{0,\Omega_i}\right\} \tag{41}$$

where $\chi \in \mathcal{S}_h$. The second term on the right hand side of (41) is handled by a standard duality estimate (Aubin-Nitsche Lemma) [27, 14]

$$\|u - u_h\|_{0,\Omega_i} \leq Ch\|\!|u - u_h|\!\|_{\Omega_i} \leq Ch\|\!|u|\!\|_{\Omega_i}.$$

For the first term, we begin with the standard approximation estimate [27,14]

$$\inf_{\chi\in\mathcal{S}_h} \|\!|u - \chi|\!\|_{\Omega_{\hat{d}}} \leq Ch|u|_{2,\Omega_{\hat{d}}}.$$

We next use the interior regularity estimate for the harmonic function u (c.f. [17]) to obtain

$$|u|_{2,\Omega_{\hat{d}}} \leq Cd^{-1}\|u\|_{1,\Omega_i}.$$

Combining these estimates, we finally have

$$\|\!|u - u_h|\!\|_{\Omega_d} \leq C\frac{h}{d}\|\!|u|\!\|_{\Omega_i}.$$

Applying this same approach to $\|\!|u - u_H|\!\|_{\Omega_d}$ yields the analogous estimate

$$\|\!|u - u_H|\!\|_{\Omega_d} \leq C\frac{H}{d}\|\!|u|\!\|_{\Omega_i}.$$

Finally, by the triangle inequality and Theorem 1

$$\|\!|u_h - u_H|\!\|_{\Omega_d} \leq \|\!|u - u_h|\!\|_{\Omega_d} + \|\!|u - u_H|\!\|_{\Omega_d} \leq C\frac{H}{d}\|\!|u|\!\|_{\Omega_i} \leq C\frac{H}{d}\|\!|u_h|\!\|_{\Omega_i}.$$

4.2 A global error estimate

Theorem 3 *Let $\hat{G} = -T(WW^t - I)T$ as in (28). Then*

$$\|T(WW^t - I)T\|_{\ell_2} \leq C\frac{H^2}{d^2} \tag{42}$$

where $C = C(\delta_0, \delta_1)$.

Proof Since $\hat{F}_i$ is symmetric, positive semi-definite,

$$\begin{aligned}
\|\hat{F}_i(I + \hat{F}_i)^{-1}\|_{\ell_2} &= \max_x \frac{x^t\hat{F}_i x}{x^t(I + \hat{F}_i)x} \\
&= \max_x \frac{x^t(S_0^{-1/2}F_iS_0^{-1/2})x}{x^t(I + S_0^{-1/2}F_iS_0^{-1/2})x} \\
&= \max_x \frac{x^tF_ix}{x^t(S_0 + F_i)x} \\
&= \max_x \frac{x^tB_i\pi_iA_i^{-1}B_i^tx}{x^t(B_iA_i^{-1}B_i^t + \sum_{j\neq i}\bar{B}_j\bar{A}_j^{-1}\bar{B}_j^t)x}.
\end{aligned} \tag{43}$$

Let $y_i = A_i^{-1}B_i^tx$. Then

$$x^tB_iA_i^{-1}B_i^tx = y_i^tA_iy_i = \|\!|\chi_h|\!\|_{\Omega_i}^2$$

for the corresponding $\chi_h \in \prod_{k=1}^p \mathcal{S}_h(\Omega_k) \equiv \mathcal{S}$. Note that the global function χ_h is determined by solving *local* problems in each of the

subdomains Ω_k, $1 \le k \le p$; these problems all have related Neumann data as specified from $B_k^t x$. Also

$$x^t B_i \pi_i A_i^{-1} B_i^t x = y_i^t A_i \pi_i y_i = |\!|\!| \chi_h - \chi_H |\!|\!|_{\Omega_i}^2$$

where $\chi_H \in \prod_{k=1}^p \mathcal{S}_H(\Omega_k) \equiv \mathcal{S}^0$ is the (piecewise elliptic) projection of χ_h into $\mathcal{S}^0$. As with χ_h, the projection χ_H is computed *locally* in each subdomain. Now let $y_j = \bar{A}_j^{-1} \bar{B}_j^t x$ for $j \ne i$. Then

$$x^t \bar{B}_j \bar{A}_j^{-1} \bar{B}_j^t x = y_j^t \bar{A}_j y_j = |\!|\!| \chi_H |\!|\!|_{\Omega_j}^2.$$

Returning now to (43),

$$\|\hat{F}_i (I + \hat{F}_i)^{-1}\|_{\ell_2} = \max_{\chi_h} \frac{|\!|\!| \chi_h - \chi_H |\!|\!|_{\Omega_i}^2}{|\!|\!| \chi_h |\!|\!|_{\Omega_i}^2 + \sum\limits_{j \ne i} |\!|\!| \chi_H |\!|\!|_{\Omega_j}^2} \tag{44}$$

Using Theorem (1),

$$|\!|\!| \chi_h |\!|\!|_{\Omega_i}^2 + \sum_{j \ne i} |\!|\!| \chi_H |\!|\!|_{\Omega_j}^2 \ge C |\!|\!| \chi_h |\!|\!|_{\Omega}^2,$$

and using Theorem (2),

$$|\!|\!| \chi_h - \chi_H |\!|\!|_{\Omega_i}^2 \le C \frac{H^2}{d^2} |\!|\!| \chi_h |\!|\!|_{\Omega_i}^2.$$

Thus, we get

$$\|\hat{F}_i (I + \hat{F}_i)^{-1}\|_{\ell_2} \le C \frac{H^2}{d^2}. \tag{45}$$

Now we turn to the main estimate. We have

$$T(WW^t - I)T = (TW)(TW)^t - T^2.$$

We will bound each of these terms separately. First, based on the definition of T and estimate (45) we have,

$$\|T^2\|_{\ell_2} = \max_i \|\hat{F}_i (I + \hat{F}_i)^{-1}\|_{\ell_2} \le C \frac{H^2}{d^2}. \tag{46}$$

Second,

$$\begin{aligned}
\|(TW)^t\|_{\ell_2}^2 &= \|TW\|_{\ell_2}^2 \\
&= \sup_x \frac{x^t W^t T^2 W x}{x^t x} \\
&= \sup_x \frac{\sum\limits_{i=1}^p x^t \hat{F}_i (I + \hat{F}_i)^{-1} x}{x^t x} \\
&= \sup_x \frac{\sum\limits_{i=1}^p x^t S_0^{\frac{1}{2}} \hat{F}_i (I + \hat{F}_i)^{-1} S_0^{\frac{1}{2}} x}{x^t S_0 x}.
\end{aligned}$$

Now use the fact that

$$y^t \hat{F}_i (I + \hat{F}_i)^{-1} y \leq y^t \hat{F}_i y,$$

which implies

$$x^t S_0^{\frac{1}{2}} \hat{F}_i (I + \hat{F}_i)^{-1} S_0^{\frac{1}{2}} x \leq x^t S_0^{\frac{1}{2}} \hat{F}_i S_0^{\frac{1}{2}} x = x^t F_i x.$$

Therefore, using Theorems 1 and 2,

$$\begin{aligned}
\|(TW)^t\|_{\ell_2}^2 &\leq \sup_x \sum_{i=1}^p \frac{x^t F_i x}{x^t S_0 x} \\
&= \sup_x \sum_{i=1}^p \frac{(B_i^t x)^t \pi_i A_i^{-1} B_i^t x}{\sum_{j=1}^p (B_j^t x)^t \bar{A}_j^{-1} B_j^t x} \\
&= \sup_{\chi_h} \sum_{i=1}^p \frac{\|\chi_h - \chi_H\|_{\Omega_i}^2}{\sum_{j=1}^p \|\chi_H\|_{\Omega_j}^2} \\
&\leq C \frac{H^2}{d^2} \sup_{\chi_h} \frac{\|\chi_h\|_{\Omega}^2}{\|\chi_H\|_{\Omega}^2} \\
&\leq C \frac{H^2}{d^2}.
\end{aligned}$$

Combining this estimate with (46) proves (42).

We can interpret (42) as follows. Certainly by forcing H/d to be sufficiently small, we can directly control the rate of convergence. As a practical matter, as p increases the coarse mesh used for load balancing in the original paradigm should become finer. In the variant strategy, the coarse mesh revealed in Step II will also naturally become finer. In either case, the natural relationship is $d \sim H$. Thus we should have $H/d \sim$ constant, which by (42) corresponds to an observed rate of convergence independent of both N and p.

5 Implementation

In this section we describe the implementation of the domain decomposition algorithm used to solve the global conforming linear systems arising in Step 3 of the Bank-Holst paradigm. Here we again use matrix notation, and note that the practical implementation differs in some respects from the idealized version of the algorithm described

in Sections 2–4. In this context, we consider the block 4×4 global saddle point problem given by

$$\begin{pmatrix} A_{ss} & A_{sm} & A_{si} & I \\ A_{ms} & A_{mm} & A_{mi} & -Z^t \\ A_{is} & A_{im} & A_{ii} & 0 \\ I & -Z & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta U_s \\ \delta U_m \\ \delta U_i \\ \Lambda \end{pmatrix} = \begin{pmatrix} R_s \\ R_m \\ R_i \\ ZU_m - U_s \end{pmatrix}. \tag{47}$$

Note that the blocking is now quite different from that used in Section 3. Here U_s refers to slave interface variables, U_m to master interface variables, U_i to subregion interior variables, and Λ to the Lagrange multipliers. The matrix A_{ii} can be ordered by subregion and is block diagonal for such an ordering. Since several slave variables can be equated to a single master variable at cross points, the matrix Z will not generally be an identity matrix; however, each row of Z will be zero except for a single entry of one corresponding to a master variable.

We formally reorder (47) as

$$\begin{pmatrix} A_{ss} & I & A_{sm} & A_{si} \\ I & 0 & -Z & 0 \\ A_{ms} & -Z^t & A_{mm} & A_{mi} \\ A_{is} & 0 & A_{im} & A_{ii} \end{pmatrix} \begin{pmatrix} \delta U_s \\ \Lambda \\ \delta U_m \\ \delta U_i \end{pmatrix} = \begin{pmatrix} R_s \\ ZU_m - U_s \\ R_m \\ R_i \end{pmatrix}. \tag{48}$$

Block elimination of the slave variables and Lagrange multipliers leads to the reduced system

$$\begin{pmatrix} A_{mm} + A_{ms}Z + Z^tA_{sm} + Z^tA_{ss}Z & A_{mi} + Z^tA_{si} \\ A_{im} + A_{is}Z & A_{ii} \end{pmatrix} \begin{pmatrix} \delta U_m \\ \delta U_i \end{pmatrix} = \begin{pmatrix} R_m + Z^tR_s - (A_{ms} + Z^tA_{ss})(ZU_m - U_s) \\ R_i - A_{is}(ZU_m - U_s) \end{pmatrix}. \tag{49}$$

The matrix appearing on the left-hand-side of (49) is the global stiffness matrix corresponding to the conforming finite element approximation. The term $R_m + Z^tR_s$ appearing on the right-hand-side corresponds to the usual residual for the conforming finite element approximation, and is independent of the choice of slave and master variables. However, the "jump" terms involving $ZU_m - U_s$ on the right-hand-side of (49) do depend on the choice of master and slave variables.

We now consider the situation on a single processor, which we denote as processor k, $1 \le k \le p$. We begin with a saddle point

problem on subregion k similar in structure to the global saddle point problem. This problem has the form

$$\begin{pmatrix} \bar{A}_{ss} & \bar{A}_{sm} & \bar{A}_{si} & I \\ \bar{A}_{ms} & \bar{A}_{mm} & \bar{A}_{mi} & -\bar{Z}^t \\ \bar{A}_{is} & \bar{A}_{im} & \bar{A}_{ii} & 0 \\ I & -\bar{Z} & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta\bar{U}_s \\ \delta\bar{U}_m \\ \delta\bar{U}_i \\ \Lambda \end{pmatrix} = \begin{pmatrix} \bar{R}_s \\ \bar{R}_m \\ \bar{R}_i \\ \bar{Z}\bar{U}_m - \bar{U}_s \end{pmatrix}. \qquad (50)$$

Here the barred quantities refer to matrices and vectors for the local problem on subregion k. For example, the block diagonal matrix $\bar{A}_{ii}$ corresponds to the interior parts of the problem on processor k. The diagonal block $\bar{A}_{ii}$ arising from region k is exactly the same as in the global saddle point problem (47). Since the remaining blocks correspond to coarse meshes, the overall order of $\bar{A}_{ii}$ is typically much smaller than A_{ii}. The residual $\bar{R}_i$ appearing on the right-hand-side of (50) has an interesting structure; for points lying in subregion k, it is the residual for the corresponding point in the global saddle point problem, and can be computed without communication on processor k. For points in the interior of $p-1$ coarse subregions, we set the residual to zero. If the local problems were all solved exactly, then the residual for the interior points would always be zero. In our case, we solve the local problems using the algebraic multilevel (multigraph) iterative method [11]. Thus, while the interior residuals will not be zero, we expect them to be much smaller than the residuals at the interface. By approximating interior residuals by zero in the coarse subregions, we avoid the need to communicate the interior residual values and to restrict them to the coarse mesh.

The interface equations are more interesting. An especially important point to emphasize here is that the designation of master and slave variables differs on each processor. The parts of the interface that involve subregion k correspond exactly to the global saddle point problem; this is of course the most crucial point. The interface unknowns associated with subregion k are all designated as the master unknowns for their mesh points, since they must be computed and updated as part of the solution process on processor k. The remaining interface points, lying on the interface of two or more subregions other than k form a subset of the interface points of the global system. For these points we define the master and slave unknowns in an arbitrary fashion (in our code, we actually use an average).

The residuals $\bar{R}_m$ and $\bar{R}_s$ can be computed using the information contained in R_m and R_s in (47) under the assumption that residuals at interior points of the global fine mesh are all zero. (Note that calculating an entry of $\bar{R}_m$ or $\bar{R}_s$ at a coarse interface point involves

expressing a coarse mesh residual as a linear combination of fine mesh residuals.) Also, the interface solution vectors $\bar{U}_m$ and $\bar{U}_s$ contain of subset of the values in U_m and U_s in (47). The parts of R_m and R_s corresponding to subregion k are computed on processor k, and processor k sends these residuals and the parts of U_m and U_s corresponding to subregion k to all other processors. In turn, processor k receives all other fine grid interface residuals and interface solution values from all other processors. This is accomplished in an *all gather* exchange in MPI. Following this exchange, each processor has all the values in R_s, R_m, U_s and U_m, and from this information can extract the subset of information needed to form $\bar{R}_s$, $\bar{R}_m$, $\bar{U}_s$ and $\bar{U}_m$.

Block elimination of the slave variables and Lagrange multipliers in (50) leads to the reduced system

$$\begin{pmatrix} \bar{A}_{mm} + \bar{A}_{ms}\bar{Z} + \bar{Z}^t\bar{A}_{sm} + \bar{Z}^t\bar{A}_{ss}\bar{Z} & \bar{A}_{mi} + \bar{Z}^t\bar{A}_{si} \\ \bar{A}_{im} + \bar{A}_{is}\bar{Z} & \bar{A}_{ii} \end{pmatrix} \begin{pmatrix} \delta\bar{U}_m \\ \delta\bar{U}_i \end{pmatrix} = \\ \begin{pmatrix} \bar{R}_m + \bar{Z}^t\bar{R}_s - (\bar{A}_{ms} + \bar{Z}^t\bar{A}_{ss})(\bar{Z}\bar{U}_m - \bar{U}_s) \\ \bar{R}_i - \bar{A}_{is}(\bar{Z}\bar{U}_m - \bar{U}_s) \end{pmatrix}. \quad (51)$$

The system matrix appearing on the left hand side of (51) is the matrix used in the final adaptive refinement step on processor k, fine in subregion k and coarse elsewhere, with possible modifications due to global fine mesh regularization. The right-hand-side can be computed once the exchange of interface data is complete. After the local system (51) is solved, the parts of $\delta\bar{U}_m$ and $\delta\bar{U}_i$ that correspond to subregion k are extracted from the solution and used to update the global solution.

We note that the choice of master and slave unknowns for points on the coarse parts of the interface on processor k is arbitrary. To resolve this ambiguity, in practice we take the master variable to be the *average* of all values that correspond to the interface point:

$$U_{i_m} \equiv \frac{1}{\ell}\sum_{s=1}^{\ell} U_{i_s}.$$

This is easy to do algorithmically, but awkward to describe in matrix notation. The effect is that the jump terms on the right-hand-side of (51) corresponding to coarse interface points are averaged over all choices of master variable. However, recall that for the interface points for subregion k, the master variable is always chosen to be the value from subregion k.

To summarize, a single domain decomposition/multigraph iteration on processor k consists of:

1. locally compute $\bar{R}_i$ and parts of R_s and R_m associated with subregion k.
2. exchange boundary data, obtaining the complete fine mesh interface vectors R_m, R_s, U_m and U_s.
3. locally compute the right-hand-side of (51) (using averages as described above).
4. locally solve (51) via the multigraph iteration.
5. update the fine grid solution for subregion k using the appropriate parts of $\delta\bar{U}_i$ and $\delta\bar{U}_m$.

We close this section with some discussion of convergence criteria. This is a delicate issue, and there are several points to consider. First, in each DD iteration each processor (simultaneously) solves the largest linear system that is solved on that processor at any point during the calculation. Although these problems might be small in comparison with the size of the global system of linear equations, they still represent the most costly calculation in the entire adaptive procedure. Second, typically we have a very good initial guess given by (6). Third, the goal of the computation is to compute an approximate solution to the PDE, not an approximate solution to the linear system (of course the two are clearly related). Fourth, we expect to have very nonuniform adaptive meshes, and the norms used in the convergence criterion should take this into account.

We begin with a discussion of norms. Let G_i denote the diagonal entry of the mass matrix corresponding to vertex i,

$$G_i = \|\phi_i\|_{0,\Omega}^2 \equiv \int_\Omega \phi_i^2 \, dx,$$

where ϕ_i is the usual nodal basis function associated with vertex i in the mesh. $G_i = O(h_i^2)$, where h_i is some measure of the size of elements sharing vertex i. Let $\mathcal{U}$ be a grid function; then

$$\|\mathcal{U}\|_G^2 = \sum_i \mathcal{U}_i^2 G_i \tag{52}$$

With this weighting, formally $\|\mathcal{U}\|_G \sim \|u_h\|_{0,\Omega}$, where u_h is the finite element function corresponding to the grid function $\mathcal{U}$. Let $\mathcal{R}$ be a residual; then

$$\|\mathcal{R}\|_{G^{-1}}^2 = \sum_i \mathcal{R}_i^2 G_i^{-1}. \tag{53}$$

With this weighting, intuitively $\|\mathcal{R}\|_{G^{-1}}$ looks like $\|e_h\|_{2,\Omega}$, where e_h is the error in the finite element solution. This must only be formal since generally $e_h \notin \mathcal{H}^2(\Omega)$.

Norms are computed with respect to the global fine mesh; each processor computes its contribution to the global norm (the contribution from vertices in $\overline{\Omega}_i$) and then a communication step is necessary to form the global norm. The main convergence criterion is

$$\frac{\|\delta\mathcal{U}^k\|_G}{\|\mathcal{U}^k\|_G} \leq \max\left(\frac{\|\delta\mathcal{U}^0\|_G}{\|\mathcal{U}^0\|_G}, \frac{\|\nabla e_h\|_{0,\Omega}}{\|\nabla u_h\|_{0,\Omega}}\right) \times 10^{-1}. \tag{54}$$

Here $\mathcal{U}^k$ and $\delta\mathcal{U}^k$ are the global grid function and update, respectively, at iteration k, while $\|\nabla e_h\|_{0,\Omega}$ and $\|\nabla u_h\|_{0,\Omega}$ are the a posteriori error estimate and the initial solution (corresponding to grid function $\mathcal{U}^0$). In words, the iteration stops when the relative error in the solution is reduced by a factor of ten, or when the relative error in the solution of the linear system is a smaller by a factor of ten than the error in the PDE at the beginning of the iteration. The norm $\|\nabla e_h\|_{0,\Omega}$ appears instead of, e.g., $\|e_h\|_{0,\Omega}$ because it arises naturally in the context of a posteriori error estimation and it is the norm for which the strongest theoretical results are available. On the other hand, the use of different norms does introduce some inconsistency into (54). One could systematically replace $\|\cdot\|_G$ with $\|\nabla\cdot\|_{0,\Omega}$ at an increased computational cost in order to resolve the inconsistency should that prove necessary. It created no problems in the numerical experiments presented in this work. A secondary convergence criterion is

$$\frac{\|\mathcal{R}^k\|_{G^{-1}}}{\|\mathcal{R}^0\|_{G^{-1}}} \leq 10^{-2}. \tag{55}$$

Typically, (54) is satisfied before (55).

Finally, on each processor the multigraph iterative method was used to solve local problems of the form (51). The convergence for the multigraph iteration was

$$\frac{\|\overline{\mathcal{R}}^j\|_{\ell^2}}{\|\overline{\mathcal{R}}^0\|_{\ell^2}} \leq 10^{-4}. \tag{56}$$

Here $\overline{\mathcal{R}}^j$ denotes the local residual at multigraph iteration j. The choice of $\|\cdot\|_{\ell^2}$ arose because the multigraph solver was part of a standalone package for solving linear systems [2] that was incorporated into PLTMG. As an algebraic multilevel method, it had no information about the linear system beyond the matrix and right hand side, and hence no basis to choose another norm. One could of course provide additional information and use another norm if necessary. The use of the more stringent tolerance 10^{-4} in (56) was to try to insure that the approximation of the interior residuals by zero at course grid points remained valid throughout the domain decomposition iteration.

6 Numerical Results

In this section, we present some numerical results. Our examples were run on a small LINUX-based Beowulf cluster, consisting of 20 dual 1800 Athlon-CPU nodes with 2GB of memory each, a dual Athlon file server, and a 100Mbit CISCO 2950G Ethernet switch. This cluster runs the NPACI ROCKS version of LINUX (based on RedHat 7.1), and employs MPICH1.2.2 as its MPI implementation. The computational kernels of PLTMG [3,2] are written in FORTRAN; the *g77* compiler (version 2.96) was used in these experiments, invoked using the script *mpif77* and optimization flag *-O*.

In these experiments, we used PLTMG to solve the boundary value problem

$$\begin{aligned} -\Delta u &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

where Ω is a domain shaped like Lake Superior.

In our first experiment, we computed an adaptive mesh with N_p vertices on a single processor. This mesh was then broadcast to p processors, where the variant strategy of combined coarsening and refinement was used to transfer approximately $N_p/2$ vertices from outside Ω_i to inside Ω_i. The global fine mesh was then made conforming as described in [7]. Note that the adaptive strategies implemented in PLTMG allow mesh moving and other modifications that yield meshes $\mathcal{T}_i$ that generally are *not* submeshes of the global conforming mesh $\mathcal{T}$. (Of course by definition they are identical on Ω_i and $\partial\Omega_i$.) However, PLTMG does insure that the partitions remain geometrically conforming, even in the coarse parts of the domain, and in particular, that the vertices on the interface system in each $\mathcal{T}_i$ are a subset of the vertices of interface system of the global mesh $\mathcal{T}$.

In this experiment, three values of N_p (50K, 75K, and 100K), and seven values of p (2^k, $1 \le k \le 7$) were used, yielding global fine meshes ranging in size form about 100K to 6.2M unknowns. Because the meshes were adaptively created, in general they violated the quasi-uniformity assumption used in the theory. Also, the goal of relocating $N_p/2$ vertices from outside Ω_i to inside Ω_i could not always be exactly satisfied for all processors, especially in the case $p = 2$. Since our cluster had only 20 nodes, for larger values of p we simulated the behavior of a larger cluster in the usual way, by allowing nodes to have multiple processes. The solution and the load balance for the case $N_p = 100K$, $p = 32$ is shown in Figure 1.

Because our analysis suggests that the interface is especially important in our algorithm, in the process of regularizing the global fine

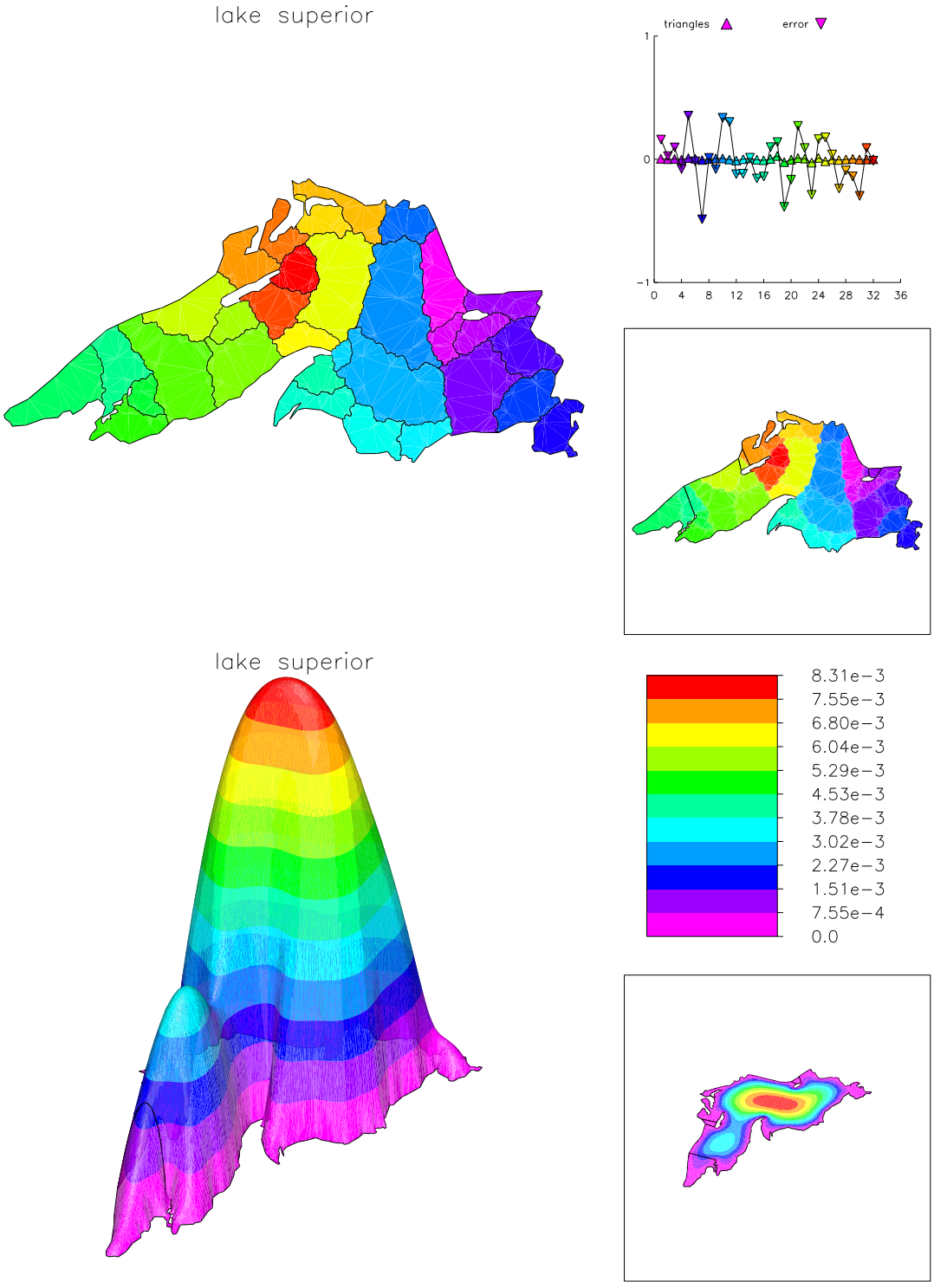

Fig. 1. The load balance (top) and solution (bottom) in the case $N_p = 100K$, $p = 32$.

mesh PLTMG also might make some additional refinement of coarse parts of the interface on each processor, simply to enhance the convergence rate of the following domain decomposition iteration. We define a graph $\mathcal{G}$ corresponding the the partition as follows. The nodes in the graph represent the subregions Ω_i, and an edge E_{ij} is present in the graph if and only if $\bar{\Omega}_i \cap \bar{\Omega}_j \neq \emptyset$. The *distance* D_{ij} is defined as the number of edges in the shortest path connecting Ω_i to Ω_j in $\mathcal{G}$. In each coarse subregion Ω_j, on the the interface $\Gamma \cap \partial\Omega_j$ we require the difference in refinement level between the local problem and the global problem to be bounded by D_{ij}. In words, the refinement on the parts of Γ outside of $\bar{\Omega}_i \cap \Gamma$ is graded in proportion to the distance in the graph $\mathcal{G}$. This strategy concentrates some additional degrees of freedom on parts of Γ where they are likely to have the greatest effect in terms of improving the rate of convergence. In this example, the amount of extra refinement for this strategy varied between none in the case $p = 2$ to less than 10% for $p = 128$. In Figure 2, we show the mesh density (local h) for the the case $N_p = 100k$, $p = 32$ for the global mesh with 1.7M vertices. We also give an example of the local mesh for one processor. Here we see the mesh coincides with the global mesh on Ω_i, and has larger elements elsewhere. We note that even with the requirement to maintain shape regularity, extra refinement along the interface is restricted to a very narrow band along the interface.

In these experiments, we modified the convergence criterion described in Section 5 to

$$\frac{\|\delta\mathcal{U}^k\|_G}{\|\mathcal{U}^k\|_G} \leq \frac{\|\delta\mathcal{U}^0\|_G}{\|\mathcal{U}^0\|_G} \times 10^{-4}. \tag{57}$$

This is more stringent than necessary for purposes of computing an approximation to the solution of the partial differential equation, but it allows us to illustrate the behavior of the solver as an iterative method for solving linear systems of equations.

Table 1 summarizes this computation. The columns labeled DD indicate the number of domain decomposition iterations required to satisfy the convergence criteria (57). For comparison, the number of iterations needed to satisfy the convergence criterion described in Section 5 is given in parentheses. From these results it is clear that the number of iterations is nearly constant over this range of N and p, despite the fact that not all assumptions of the theory were satisfied. The size of the global mesh for the variant strategy can be estimated from the formula

$$N \approx p\theta N_p + N_p \tag{58}$$

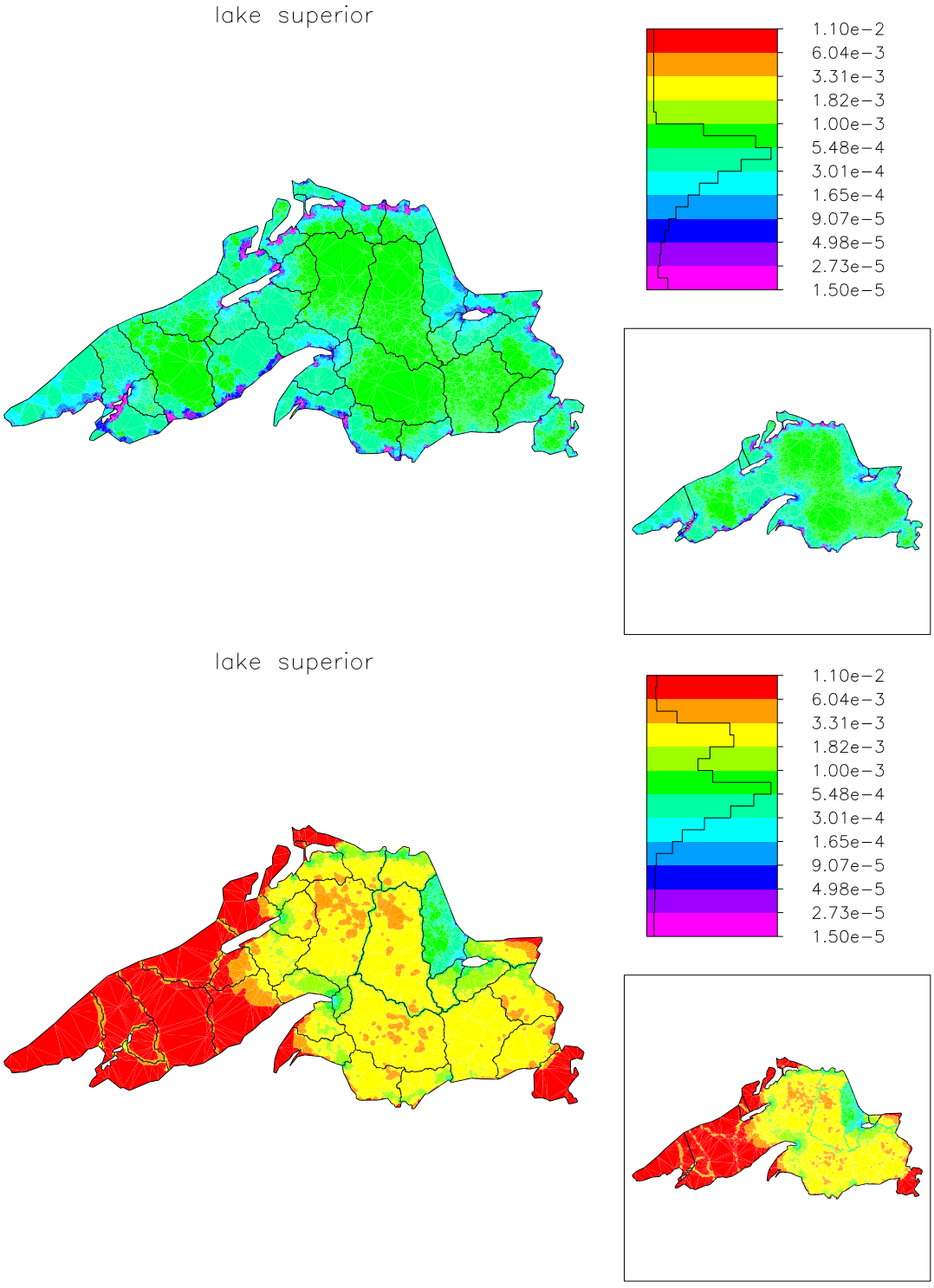

Fig. 2. The mesh density for the global mesh (top) and for one of the local meshes (bottom) in the case $N_p = 100K$, $p = 32$.

for $\theta = 1/2$. For $N_p = 100K$, $p = 128$, (58) predicts $N = 6300000$, where the observed $N = 6263743$.

Table 1. Convergence Results for Variant Algorithm. Numbers of iterations needed to satisfy (57) are given in the column labeled DD. The numbers in parentheses are the number of iterations required to satisfy the convergence criterion described in Section 5.

	$N_p = 50K$		$N_p = 75K$		$N_p = 100K$	
p	N	DD	N	DD	N	DD
2	99107	10 (3)	148938	12 (3)	198861	7 (3)
4	150108	7 (3)	225145	7 (3)	300166	6 (2)
8	249613	7 (3)	374395	7 (3)	499269	6 (2)
16	446250	7 (3)	670594	8 (3)	894626	8 (3)
32	835266	7 (3)	1257664	7 (3)	1678726	7 (3)
64	1599512	7 (3)	2413590	6 (3)	3226399	7 (3)
128	3086008	6 (3)	4675761	7 (3)	6263743	7 (3)

In our second experiment we solved the same problem using the original paradigm. On one processor, an adaptive mesh of size $N_c = 10K$ was created. This mesh was then partitioned into p subregions, $p = 2^k$, $1 \leq k \leq 7$. This coarse mesh was broadcast to p processors (simulated as needed) and each processor continued the adaptive process, creating a mesh of size N_p. We chose three values for N_p, 50K, 75K, and 100K. This resulted in global meshes varying in size from approximately 90K to 11M vertices. These global meshes were regularized, and a global DD solve was made as in the first experiment. As in the first experiment, the usual convergence criteria was replaced by (57) in order to illustrate the dependence of the convergence rate on N and p. The results are summarized in Table 2.

For the original paradigm the size of the global mesh is predicted by

$$N \approx pN_p - (p-1)N_c. \tag{59}$$

Equation (59) predicts an upper bound, as it does not account for refinement outside of Ω_i, needed to keep the mesh conforming and for other reasons. For example, for N_c=10K, N_p=100K, $p = 128$, (59) predicts $N \approx 11530000$ when actually $N = 10921132$.

References

1. I. Babuška and J. M. Melenk, *The partition of unity method*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 727–758.

Table 2. Convergence Results for Original Algorithm. Numbers of iterations needed to satisfy (57) are given in the column labeled DD. The numbers in parentheses are the number of iterations required to satisfy the convergence criterion described in Section 5.

	$N_p = 50K$		$N_p = 75K$		$N_p = 100K$	
p	N	DD	N	DD	N	DD
2	89738	6 (2)	139599	6 (2)	189408	6 (2)
4	168618	7 (3)	267831	7 (3)	367091	7 (2)
8	324674	6 (2)	522879	7 (3)	721028	7 (2)
16	630749	7 (3)	1026060	7 (3)	1421128	7 (3)
32	1231557	8 (3)	2022028	7 (3)	2811006	7 (3)
64	2397415	7 (3)	3975520	7 (3)	5551497	6 (3)
128	4614399	8 (3)	7770798	7 (3)	10921132	8 (3)

2. R. E. Bank, *Multigraph users' guide - version 1.0*, tech. report, Department of Mathematics, University of California at San Diego, 2001.
3. ———, *PLTMG: A software package for solving elliptic partial differential equations, users' guide 9.0*, tech. report, Department of Mathematics, University of California at San Diego, 2004.
4. ———, *Some variants of the Bank-Holst parallel adaptive meshing paradigm*, Computing and Visualization in Science, (accepted).
5. ———, *A domain decomposition for a parallel adaptive meshing algorithm*, in Sixteenth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Lecture Notes in Computational Science and Engineering, Springer-Verlag, to appear.
6. R. E. Bank and M. J. Holst, *A new paradigm for parallel adaptive meshing algorithms*, SIAM J. on Scientific Computing, 22 (2000), pp. 1411–1443.
7. ———, *A new paradigm for parallel adaptive meshing algorithms*, SIAM Review, 45 (2003), pp. 292–323.
8. R. E. Bank and P. K. Jimack, *A new parallel domain decomposition method for the adaptive finite element solution of elliptic partial differential equations*, Concurrency and Computation: Practice and Experience, 13 (2001), pp. 327–350.
9. R. E. Bank, P. K. Jimack, S. A. Nadeem, and S. V. Nepomnyaschikh, *A weakly overlapping domain decomposition preconditioner for the finite element solution of elliptic partial differential equations*, SIAM J. on Scientific Computing, 23 (2002), pp. 1817–1841.
10. R. E. Bank and S. Lu, *A domain decomposition solver for a parallel adaptive meshing paradigm*, SIAM J. on Scientific Computing, 26 (2004), pp. 105–127 (electronic).
11. R. E. Bank and R. K. Smith, *An algebraic multilevel multigraph algorithm*, SIAM J. on Scientific Computing, 23 (2002), pp. 1572–1592.
12. F. Belgacem, *The mortar finite element method with Lagrange multipliers*, 1997. Preprint.
13. C. Bernardi, Y. Maday, and A. Patera, *A new nonconforming approach to domain decomposition: the mortar element method*, in Nonlinear partial differential equations and their applications, H. B. adn J.L. Lions, ed., Pitman Research Notes in Mathematics, New York, 1994, John Wiley and Sons, pp. 13–51.

14. D. Braess, *Finite elements*, Cambridge University Press, Cambridge, 2001.
15. D. Braess, W. Dahmen, and C. Weiners, *A multigrid algorithm for the mortar finite element method*, SIAM J. on Numerical Analysis, 37 (1999), pp. 48–69.
16. T. Chan and T. Matthew, *Domain decomposition algorithms*, in Acta Numerica, Cambridge University Press, 1994, pp. 61–143.
17. L. C. Evans, *Partial differential equations*, vol. 19 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 1998.
18. J. Huang and J. Zou, *Construction of explicit extension operators on general finite element grids*, Appl. Num. Math., 43 (2002), pp. 211–227.
19. J. Huang and J. Zou, *A mortar element method for elliptic problems with discontinuous coefficients*, IMA J. Numer. Anal., 22 (2002), pp. 549–576.
20. S. Lu, *Parallel Adaptive Multigrid Algorithms*, PhD thesis, Department of Mathematics, University of California at San Diego, 2004.
21. J. M. Melenk and I. Babuška, *The partition of unity finite element method: basic theory and applications*, Comput. Methods Appl. Mech. Engrg., 139 (1996), pp. 289–314.
22. W. Mitchell, *The full domain partition approach to distributing adaptive grids*, Applied Numerical Mathematics, 26 (1998), pp. 265–275.
23. ———, *A parallel multigrid method using the full domain partition*, Electronic Transactions on Numerical Analysis, 6 (1998), pp. 224–233.
24. W. F. Mitchell, *The full domain partition approach to parallel adaptive refinement*, in Grid generation and adaptive algorithms (Minneapolis, MN, 1997), vol. 113 of IMA Vol. Math. Appl., New York, 1999, Springer, pp. 151–161.
25. J. A. Nitsche and A. H. Schatz, *Interior estimates for Ritz-Galerkin methods*, Math. Comp., 28 (1974), pp. 937–958.
26. B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
27. G. Strang and G. J. Fix, *An analysis of the finite element method*, Prentice-Hall Inc., Englewood Cliffs, N. J., 1973.
28. A. Toselli and O. Widlund, *Domain decomposition methods—algorithms and theory*, vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005.
29. L. B. Wahlbin, *Local behavior in finite element methods*, in Handbook of numerical analysis, Vol. II, Handb. Numer. Anal., II, North-Holland, Amsterdam, 1991, pp. 353–522.
30. C. Wieners and B. I. Wohlmuth, *Duality estimates and multigrid analysis for saddle point problems arising from mortar discretizations*, SIAM J. Sci. Comput., 24 (2003), pp. 2163–2184 (electronic).
31. B. I. Wohlmuth, *A mortar finite element method using dual spaces for the Lagrange multiplier*, SIAM J. Numer. Anal., 38 (2000), pp. 989–1012 (electronic).
32. J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review, 34 (1992), pp. 581–613.
33. J. Xu and A. Zhou, *Local and parallel finite element algorithms based on two-grid discretizations*, Math. Comp., 69 (2000), pp. 881–909.